

REMARKS

Claims 1, 3-12, 14-22, 24, and 26-29 are pending in the present application. Claims 1, 11, 12, 22, 24, and 26 are amended. Support for the amendments may be found in the specification on at least page 14, line 18, to page 15, line 3. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 1, 3-12, 14-22, 24, and 26-29 under 35 U.S.C. § 103 as being unpatentable over *Judd et al.* (U.S. Patent No. 6,360,215) in view of <http://www.Aspgrid.com/faq.html> by Persits Software, Inc. Copyright 2000, hereinafter referred to as "*AspGrid*." This rejection is respectfully traversed.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). In this particular case, the Office Action fails to establish a *prima facie* case of obviousness because the cited references do not teach or suggest all of the features of the presently claimed invention.

Judd teaches a method and apparatus for retrieving documents based on information other than document content. In *Judd*, a search engine generates an index for documents and adds tag words to the index when matches are found. Thereafter, the search engine of *Judd* executes searches against the index, rather than the actual content of the document. See *Judd*, Abstract.

In contradistinction, the present invention provides a mechanism for searching a Web page responsive to receiving a single input string, wherein the input string includes a universal resource identifier of the Web page and a search string. *Judd* does not teach or suggest parsing an input string for a universal resource identifier and a search string and searching for the information corresponding to the search string through a Web page identified by the universal resource identifier, as recited in claim 1. More particularly, *Judd* does not teach or suggest that a universal resource identifier and a search string are separated from each other in the input string by a selected delimiter, as recited in claim 1.

To the contrary, *Judd* teaches using tag words and document specifications to search Web pages. In col. 9, lines 1-35, which is cited in the Office Action as allegedly

teaching the claim limitations, *Judd* states:

A tag word is any character string that is to be associated with a document for search purposes. Often, the tag words are dedicated code words, or words that are not normally found in a document or dictionary, although this characteristic is not required. Examples of tag words include "n2h2/black" and "n2h2/white", as shown by tag words 138a, 138c of FIG. 3. Other tag words may be properties or meta-information such as the title of a document, abstract, or others, as described further below. For example, a tag word may be "ADVERTISEMENT" to indicate that its associated Web page(s) contain advertising. A tag word may be "VERIFIED" to indicate that its associated Web page(s) contain factual information that has been verified by some independent third party.

Each of the document specifications 132 specifies a matching criteria. The documents that satisfy the matching criteria are specified in a document specification associated, in the index, with the tag word 134 of the record having that document specification. A document specification 132 may, for example, an expression that identifies the location of a document in a network, such as a URL.

In a preferred embodiment, the document specification may be expressed in a wildcard format. Using a wildcard format for the document specification allows a particular tag word to be associated with more than one URL, without requiring each URL to be identified literally. Document specifications 136a-136c of FIG. 3 are expressed in wildcard format. For example, document specification 136a is `http://*.hotsex.com/`. The "*" character in document specification 136a indicates that the document specification includes any server within the domain "hotsex.com". When the process of FIG. 2 processes document specification 136a, the code word 138a will be associated in the index 16 with any indexed document having a location identifier that matches document specification 136a, as explained further below.

There is no teaching in the cited portion, or any other portion, of *Judd* of parsing an input string for a universal resource identifier and a search string. The Office Action proffers no analysis as to why tag words and document specifications are somehow equivalent to an input string that includes a universal resource identifier and a search string. The Office Action also fails to explain how *Judd* somehow teaches searching for the

information corresponding to the search string through a Web page identified by the universal resource identifier that is provided in the input string, as recited in claim 1.

Rather, *Judd* teaches receiving a typical search string (query) that contains one or more words. See col. 8, lines 22-29. There is no teaching whatsoever in *Judd* of an input string in which a universal resource identifier and a search string are separated from each other by a selected delimiter, as recited in claim 1. The Office Action provides no analysis as to why *Judd* somehow teaches this feature other than to cite a seemingly arbitrary portion of the reference that describes a wildcard being used in a URL in a document specification.

The Office Action cites *AspGrid* as teaching parsing the input string for a universal resource identifier and a search string, wherein the universal resource identifier and the search string are separated from each other by a selected delimiter. The Office Action notes that if you enter a search of "string," for example, into the Web site <http://www.aspgrid.com>, the result page will have a URL as follows:

<http://www.aspgrid.com/search.asp?ciRestriction=string>

However, the Office Action does not explain how this somehow anticipates a step of parsing the string for a selected delimiter.

Claim 1, as amended, recites a step of determining whether the Web page associated with the uniform resource identifier has a search object. If the Web page has a search object, then the search is performed using the search object. However, if the Web page does not have a search object, the present invention performs a simple text search of the Web page. Neither *Judd* nor *AspGrid* teaches or suggest determining whether the Web page has a search object. In the *AspGrid* example proposed in the Office Action, the Web page is presumed to have a search object; otherwise, the string "search.asp?ciRestriction=" would have no meaning.

Furthermore, neither *Judd* nor *AspGrid* teaches or suggests performing a simple text search if the Web page does not have a search object. In the *AspGrid* example proposed in the Office Action, the search object is deliberately invoked. In fact, *AspGrid* teaches away from the present invention, because *AspGrid* teaches the user may deliberately invoke an Application Server Page that has a search object, as opposed to

determining whether a search object exists and performing a search using the search object or performing a simple text search based on the determination.

Independent claims 11, 12, 22, 24, and 26-29 recite subject matter addressed above with respect to claim 1 and are allowable for at least the same reasons. Since claims 3-10 and 14-21 depend from claims 1 and 12, the same distinctions between *Judd* and *AspGrid* and the invention recited in claims 1 and 12 apply for these claims. In addition, claims 3-11, 14-22, 24, and 26-29 recite additional combinations of features not taught or suggested by the applied references.

With respect to claims 10 and 21, the Office Action states:

Regarding claims 10 and 21, most of the limitations of these claims have been noted in the rejection of claims 1 and 12 above, respectively. In addition, *Judd/Persist Software* discloses: wherein the selected delimiter is at least one of a "\$", "o", "*", and "#" (col. 10, lines 60 to col. 11, lines 61, *Judd*).

Office Action, dated September 3, 2004. Applicants respectfully disagree. The cited portion of *Judd* describes a wildcard format for document specifications. Nowhere in the cited portion of *Judd*, or any portion of either applied reference, does the prior art teach a delimiter for separating a universal resource identifier and a search string. The Office Action proffers no analysis whatsoever as to why a wildcard in a document specification is somehow equivalent to a delimiter in an input string, wherein the delimiter separates a universal resource identifier and a search string, as in the claimed invention. *AspGrid* does not solve the deficiencies of *Judd*. The applied references, taken alone or in combination fail to teach or suggest each and every claim limitation. Therefore, the proposed combination of *Judd* and *AspGrid* cannot render claims 10 and 21 obvious.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1, 3-12, 14-22, 24 and 26-29 under 35 U.S.C. § 103.

II. Conclusion

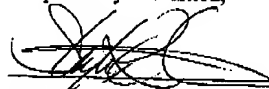
It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE:

December 3, 2007

Respectfully submitted,



Stephen R. Tkacs
Reg. No. 46,430
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicants